Not an Aho-Corasick Problem

Time limit: 3.0s **Memory limit:** 512M

Note: this problem is solvable *without* the use of the Aho-Corasick algorithm.

The Aho-Corasick algorithm is a well ~~known~~ hated algorithm used to solve many ~~fun~~ annoying string problems. Wesley was recently asked to create a problem involving the Aho-Corasick algorithm. After taking one look at an implementation, he immediately declined. In fact, he was so disgusted by it that he was inspired to make a problem that did not involve the use of the Aho-Corasick algorithm.

However, even problems that do not use the Aho-Corasick algorithm can seem like they do. A very recent ECOO problem involving words in the English language seemed to require the Aho-Corasick algorithm, and not wanting to mislead people into thinking they needed this ~~intriguing~~ disgusting algorithm, Wesley wanted to make it very clear that the **Aho-Corasick algorithm will not be required to solve this problem**.

You recently found a dictionary containing N words. Please note that this dictionary should not be used with the Aho-Corasick algorithm. Unlike the ECOO problem, this dictionary may contain words that are not in the English language, or any spoken language. Do not fear though, as this problem will not involve the use of the Aho-Corasick algorithm. Wesley found this dictionary to be very boring, and decided to create a new language by concatenating rotations of strings in the dictionary. This new language has an infinite number of words (as long as there is at least one word in the original dictionary).

Formally, let the strings in the dictionary be $S_1, S_2, \ldots, S_N$. Pick a positive integer K. Select K integers: $A_1, A_2, \ldots, A_K$, where $1 \leq A_i \leq N$ for all i. For each i ($1 \leq i \leq K$), let T_i be a rotation of the string S_{A_i}. Let the string Z be the concatenation of $T_1, T_2, \ldots, T_N$. The string Z is part of the new language.

For example, if the dictionary contains the words `aho` and `corasick`, words in the new language include (but are not limited to): `aho`, `hoa`, `oah`, `corasick`, `ahocorasick`, `hoaaho`, `sickcorahoa`, and `oahickcorasckcorasihoa`. Words that are **not** in the new language include (but are not limited to): `oha`, `hao`, and `acorahsicko`. To be clear, even if the word `ahocorasick` is in the new language, the Aho-Corasick algorithm does not need to be used to solve the problem.

Given a string X, Wesley wants to know how many different ways there are to form this new word from the dictionary you found. Since this number can be very large, you should output it modulo a number M, that will be provided in the input. Please see the output specifications for more details. **Please note that under no circumstance will you be required to run the Aho-Corasick algorithm on X.**

Constraints

$1 \leq N \leq 5 \cdot 10^4$

$1 \leq M \leq 10^9$

$1 \leq |X| \leq 5 \cdot 10^4$

The sum of all $|S_i|$ is at least 1 and no greater than 10^5.

Each string S_i consists only of lowercase letters in the English alphabet (but may or may not be English words).

In addition, the problem is guaranteed to be solvable without the Aho-Corasick algorithm.

Input Specification

The first line contains 2 integers N and M, subject to the constraints above.

The next N lines describe the dictionary. Each line contains a single string S_i, subject to the constraints above.

The last line contains the string X, subject to the constraints above.

Output Specification

This problem is graded with an `identical` checker. This includes whitespace characters. Ensure that every line of output is terminated with a `\n` character.

~~Output 0 if you used the Aho-Corasick algorithm to solve this problem, and 1 otherwise. Only solutions that did not use the Aho-Corasick algorithm are considered correct.~~

Output a single integer, the number of different ways to form this new word from the dictionary. Two ways are considered different if X is formed by concatenating a different rotation of strings.

Formally, let one way of forming X be concatenating the strings $T_1, T_2, \ldots, T_K$, where T_i is a rotation of the string S_{A_i}, $1 \leq A_i \leq N, 1 \leq i \leq K$. Let another way of forming X be by concatenating the strings $U_1, U_2, \ldots, U_L$ where U_i is a rotation of the string S_{B_i}, $1 \leq B_i \leq N, 1 \leq i \leq L$. Two ways are considered different if $K \neq L$ or if there exists an i where $A_i \neq B_i$. **Different rotations of the same string T_i are not considered different ways.**

Note: users who submit a solution using the Aho-Corasick algorithm may be banned from making further submissions to this problem ... okay maybe not.

Sample Input 1

```
2 2
sickcora
hoa
ahocorasick
```

Sample Output 1

```
1
```

Sample Explanation 1

~~Since the Aho-Corasick algorithm was not used to solve this problem, the answer is 1.~~

The only way to make the word `ahocorasick` is by concatenating the second word in the dictionary, rotated once to the right, and the first word, rotated 4 times to the right. $1 \bmod 2 \equiv 1$.

Sample Input 2

```
5 3
cab
abc
cba
aa
bc
bcaabc
```

Sample Output 2

```
2
```

Sample Explanation 2

Using the formal definition of a unique way to form X from the output statement above, the 5 unique arrays A are $\{1, 1\}$, $\{2, 2\}$, $\{1, 2\}$, $\{2, 1\}$, and $\{5, 4, 5\}$. $5 \bmod 3 \equiv 2$.

Sample Input 3

```
2 5
wx
yz
wyxz
```

Sample Output 3

```
0
```

Sample Explanation 3

There are no ways to make the string `wyzx` by concatenating rotations of words in the dictionary. $0 \bmod 5 \equiv 0$.